Applying Tools and Techniques of Natural Language Processing to the Creation of Resources for Less Commonly Taught Languages

Marjorie McShane
University of Maryland, Baltimore County

Abstract

This paper proposes that research results from the area of natural language processing could effectively be applied to creating software to facilitate the development of language learning materials for any natural language. We will suggest that a knowledge-elicitation system called Boas, which was originally created to support a machine-translation application, could be modified to support language-learning ends. Boas leads a speaker of any natural language, who is not necessarily trained in linguistics, through a series of pedagogically-supported questionnaires, the responses to which constitute a "profile" of the language. This profile includes morphological, lexical and syntactic information. Once this structured profile is created, it can feed into virtually any type of system, including one to support language learning. Creating language-learning software using a system like this would be efficient in two ways: first, it would exploit extant cutting-edge research and technologies in natural language processing; and second, it would permit a single tool to be used for all languages, including less commonly taught ones, for which limited funding for resource development is a bottleneck.

Introduction

This article is about creatively applying knowledge, methodologies, and resources developed in the field of natural language processing (NLP) to the needs of teachers and students of less commonly taught languages (LCTLs). For the most part, teachers and learners of LCTLs must make do with fewer and less advanced teaching materials than their counterparts in the more popular languages, like English, Spanish and French. Moreover, there is a smaller promise of recompense for creating such resources. Brecht and Walton (no

date) discuss this issue, proposing that we need a "Language Learning Framework" that can "guide the design and management of instructional programs, materials development, teacher training, standards and assessment systems, and the whole range of infrastructure components [...] upon which individual teachers and programs depend." One way to speed the development of teaching and learning resources, at relatively minimal cost, is to adapt available, parametrizable resources to one's own needs. Although adaptation may pose some challenges that would not arise in a custom-built system, using a configurable system also offers methodologies and insights drawn from the common wisdom.

This paper focuses on the conceptual, rather than technological, aspects of a knowledge-elicitation (KE) system called Boas. This system, named after renowned field linguist Franz Boas, elicits knowledge about any natural language (L) in an organized, methodologically sound and pedagogically supported way, resulting in a language profile that can be used for many purposes, including language learning. There are two obvious types of language-learning applications for such a profile: 1) the development of a grammar that can be printed out or accessed on-line and 2) the development of interactive exercises and other study materials drawing on that grammar.

Boas: the Seed System

Boas was originally developed to elicit knowledge to support the creation of systems that translate from any language, L, into English. The idea was to present a speaker of L with a translation system that lacked only one component: information about L. That is, upon delivery to a language informant, the system already contains a grammar and lexicon of English, machine-translation engines, and a knowledge-elicitation (KE) component, which elicits all the information about L needed to configure the machine translation system. Once the user provides that information, he or she pushes a button and gets a moderate-quality translation system.

Boas and the larger system that houses it, Expedition, were built primarily for so-called "low density" languages-those for which few or no resources are available. In pedagogical terms, these correspond to the much less commonly, least commonly, and rarely or never taught languages (Brecht and Walton). There are practical reasons for this focus: although having some machine translation capabilities for such languages is far better than having none-at least in the realm for which the project was contracted-the quality of a translation system generated in template form cannot compete with that of a system cater-made for a given language pair. Thus, Boas intends to fill a very specific niche.

Not only the target languages of the system but also the "rules of the game" for employing it derive from practical considerations. Since there may be no trained linguist for a given language who would be available to work as an informant, the system must be accessible to naïve informants - who must, however, know both L and English well. Similarly, since informant time is a costly resource, the knowledge-elicitation process should take only about six months of work by a single informant.

The knowledge-elicitation component developed for Boas represents an innovative methodology of knowledge elicitation, which is what makes the system accessible even to linguistically novice informants, permits it to cover any natural language, and allows its incremental extension as resources become available or the scope of interest expands.

The KE process is based upon our understanding - derived of cross-linguistic research - of what phenomena occur in language and, tangentially, our view of what needs to be covered to describe a language to a reasonable degree of detail. (The latter can, of course, be reevaluated based upon a given application, be that application a machine translation system or a language course.) We organize "what can occur in language" into a series of parameters, their value sets, and their means of realization, as shown by the samples in Table 1. The first block illustrates inflection, the second, closed-class lexical meanings, the third, "ecology" (the inventory of characters in L, the expression of dates, numbers, etc.), and the fourth, syntax.

Parameter	Values	Means of Realization
Case Relations	Nominative, Accusative, Dative, Instrumental, Abessive, etc.	flective morphology, agglutinating morphology, isolating morphology, prepositions, postpositions, etc.
Number	Singular, Plural, Dual, Trial, Paucal	flective morphology, agglutinating morphology, isolating morphology, particles, etc.
Tense	Present, Past, Future, Timeless	flective morphology, agglutinating morphology, isolating morphology, etc.
Possession	+/-	case-marking, closed-class affix, word or phrase, word order, etc.
Spatial Relations	above, below, through, etc.	word, phrase, preposition or postposition, case-marking
Expression of Numbers	integers, decimals, percentages, fractions, etc.	numerals in L, digits, punctuation marks (commas, periods, percent signs, etc.) or a lack thereof in various places
Sentence Boundary	declarative, interrogative, imperative, etc.	period, question mark(s), exclamation point(s), ellipsis, etc.
Grammatical Role	subjectness, direct-objectness, indirect-objectness, etc.	case-marking, word order, particles, etc.
Agreement (for pairs of elements)	+/- person, +/- number, +/- case, etc.	flective, agglutinating or isolating inflectional markers

Table 1 Sample parameters, values and means of their realization.

Although Boas currently covers a large inventory of parameters, values and means of realization, our lists are sure to be incomplete, which is why all inventories are supplemented with the option "add a new parameter/value". For example, if nouns in L inflect for the

parameter Case but some needed value of Case is missing from the inventory, it may be typed in a text field and processed in the same way as all the other cases. This facility prepares the system to cover most phenomena in most languages.

The methodology used in Boas weds system- and user-initiative. The KE process is organized as a series of (sub) tasks, with the order of work restricted only inasmuch as prerequisites for certain tasks obtain; apart from those restrictions, work can proceed in any order. The tasks are presented to the informant in a dynamic task tree supplemented with icons that indicate task status. Figure 1 shows the task tree at the point when the paradigmatic morphology of nouns is being started. The "green light" icon shows that the task Introduction can be accessed. The "do not enter" icons show that the tasks below it have prerequisites and cannot currently be accessed.

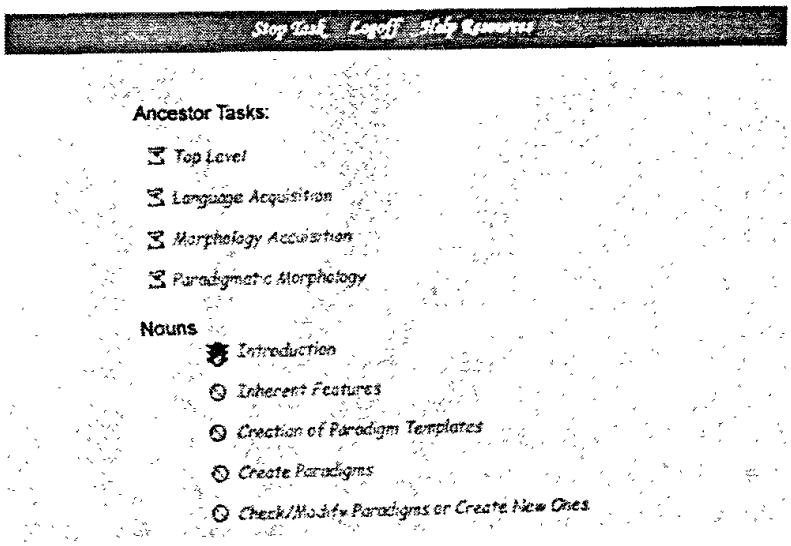


Figure 1

Boas caters to users of different levels of linguistic experience through methods of progressive disclosure, by which support information is provided as needed. Figure 2 shows two means of progressive disclosure: 1) in the lower left-hand corner are three hyperlinks that provide information on the stated topics; the page accessed from the second one in shown in Figure 3; 2) the Help Resources link in the always-available blue frame leads to glossaries, tutorials, and what amounts to an on-line textbook of descriptive linguistics that was written expressly for this system.

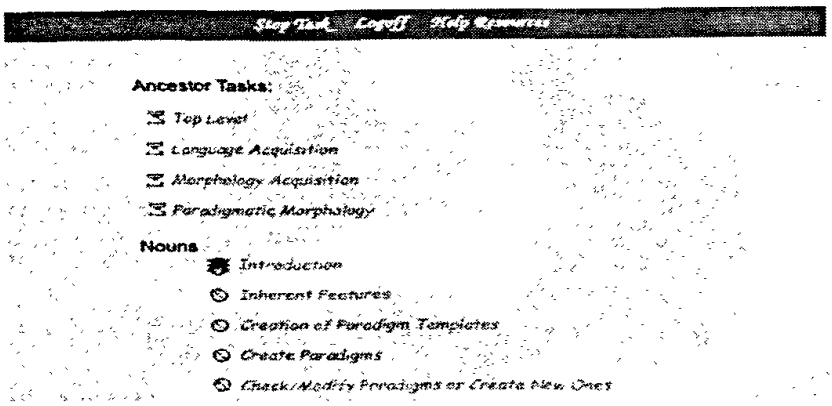


Figure 2

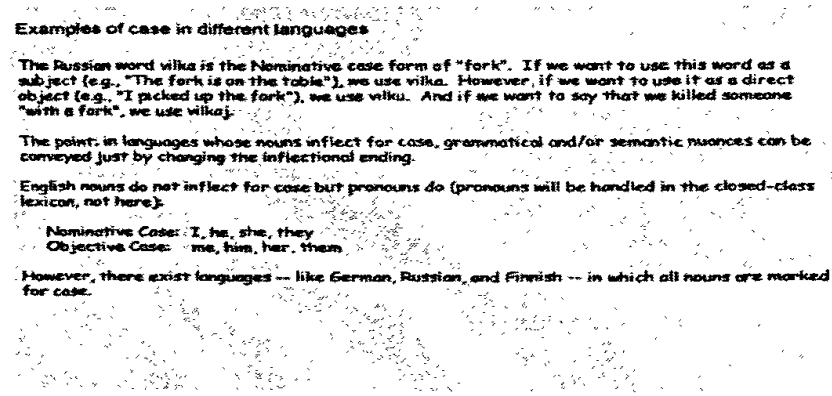


Figure 3

The architecture of this system has many advantages. First, it permits tasks to be carried out in various ways. For example, if a one can obtain an on-line L-to-English dictionary, it can be reformatted and imported into the system. If no such materials are available, the threads of knowledge acquisition can be followed from scratch. Second, tasks can always be returned to and be redone or edited. So, one might import a rudimentary on-line dictionary then expand it incrementally as desired. Third, because of its Web-based platform, resources can be shared by users. For example, if a user configures an excellent system for Yoruba, he/she can choose to make it available to select others or to the whole world. Individual users can then edit, expand or update their personal copy of the system as desired.

The Boas knowledge-elicitation system is quite separate from the machine-translation system for which it was developed. As such, it could be nested in a pedagogical system that would work as follows. A superuser-who might be a teacher of L, a graduate student, or any literate speaker of the language with reasonable analytical skills-develops a profile of L using Boas. This, in itself, would be a large step in the development of resources for less commonly taught lan-

guages. The profile could be distributed to other teachers, printed out for students as a textbook, etc. Depending on how big the lexicon will be, how many (if any) inflectional patterns must be covered, how many external resources can be imported, how comprehensive the language profile will be, etc., this process could take anywhere from a couple of weeks to several months. However, since the profile is infinitely extensible, the most time-consuming effort-building the open-class lexicon-can be carried out over any period of time. Once the language profile is complete to some grain size of description and degree of coverage, it can start to be used as a teaching/learning resource.

The Modules of Boas and Proposed Pedagogical Extensions

Although space does not permit a full description of the tasks in Boas, the brief descriptions below should suffice for purposes of orientation. Sample pedagogical extensions are proposed, which represent only a glimpse into what a Boas for LCTLs (hereafter, LCTL-Boas) might ultimately look like.

Ecology

Ecology is a term used in natural language processing to describe those features of language that lie outside of traditional grammar and lexis. The ecology module of Boas elicits the inventory of characters used in L (and their division into vowel, consonant and "other"), the inventory and use of punctuation marks, proper name conventions and means of expressing dates and numbers. Although these top-level aspects of texts are not traditionally organized into a separate topic of study, they must be learned by any person or machine attempting to read texts in L.

Morphology

The morphology module of Boas covers inflectional and derivational morphology, which are treated separately.

Inflectional Morphology

In the module for inflectional morphology, the informant is first taught how to determine whether L has flective, agglutinating, isolating, mixed or no inflectional morphology. If all or part of inflection is best captured using paradigms, Boas guides the informant through the process of providing sample paradigms from which a morphology learning program can infer rules that are later applied to the whole open-class lexicon. The process of creating inflectional paradigms involves two steps: creating a paradigm template with any layout the user prefers and filling it with sample words. The informant can split paradigms finely or bunch them, depending upon his or her own preferences. Several aspects of this module are important for language learning.

1. An untrained user (who could even be a student studying a rare language independently) is carefully guided in creating para-

digms, and extensive explanatory materials and examples are provided.

2. Based on the user's preferences (and the selected machine-learning program), a given language can be described as having 4, 14, or 40 paradigms for a given part of speech. In fact, one could even list all inflectional forms for all words directly, circumventing reliance on machine learning, if making the kinds of linguistic generalizations required for paradigm delineation proved too difficult for some informant of some language.

3. Inflectional morphology is one realm in which computer support of teaching has been widely used, since drilling is an inevitable part of internalizing inflectional patterns. Having a full on-line inventory of paradigms whose rules can be applied to the entire open-class lexicon would greatly expand drilling possibilities.

4. The flexibility in paradigm layout permits users to select the most memorable, helpful method of displaying the paradigm.

5. One task that is always available in Boas, after initial praradigm delineation, is reviewing, editing and supplementing the inventory of paradigms. So, a while a basic inventory might suffice for beginning students, the full inventory will be required for advanced ones.

6. The collected information can be viewed and printed out using a variety of summary functions. For example, one might want to view/print out the list of nominal paradigms and their test members, or, one might want to view one of the actual paradigms followed by the list of sample members. In order to increase the efficacy of LCTL-Boas, additional features could be added, like a space to provide a prose description of and diagnostics for each paradigm. Thus, teachers and learners could advance pedagogical practice by organizing the presentation of inflection however they deem best.

Pedagogical supplements that could be added, in template fashion, to the flective module of LCTL-Boas include exercises like the following: (i) provide all inflectional forms for a selected list of words; (ii) provide all inflectional forms of random words from some paradigm; (iii) provide all inflectional forms of random words belonging to some part of speech; (iv) provide selected inflectional forms for selected words or random words from some part of

speech; (v) click on the combination of parameter values represented by some inflectional forms; (vi) group words into paradigms in a game by which the user catches words falling down the screen in a bag representing the paradigm. In all such exercises, all answers could be checked and any mistaken forms highlighted, unless the teacher would disable this option for homework assignments or testing purposes. In fact, any of the interactive drills and games that would be created for individual languages could be incorporated into generalized Boas as long as they could be recast as parametrizable templates. The Boas KE methodology would then lead teachers through the process of filling those templates with the necessary information to gear up the exercises for a particular language.

If L has agglutinating or isolating inflectional morphology, Boas collects the affixes and/or free-standing words used to realized basic grammatical meanings. For example, the Turkish word ta°ittim, which means 'I made someone carry (something)', contains a stem, ta°i 'carry', plus three agglutinating affixes: t-causitive, ti-past, and m-first singular. Agglutinating and isolating inflectional units are elicited together because 1) the prompts are the same-the inventory of parameters and values mentioned above and 2) the method of indicating them is the same-typing in one or more strings (i.e., series of characters) into a text field. The only difference is that for affixes the point of attachment must be indicated. The same type of training exercises as mentioned above could be used if patterns of inflectional morphology are provided in this way.

Derivational Morphology

Derivational morphology is a difficult aspect of grammar to generalize about because, both in terms of form and in terms of meaning, simple concantenation often does not obtain. That is, adding derivational affixes to words often causes boundary and/or word-internal spelling changes; and even if the rules for such spelling changes could be listed (which is possible for some processes in some languages), the semantics of the resulting entity would often not be predictable, as derivational affixes are often ambiguous. For example, -er in English is typically taken to be an affix that, when attached to a verb, V, produces a noun whose meaning is "the agent of V-ing." However, this analysis certainly does not apply to the word cooker. Semantic non-compositionality like this is common not only for affixal word formation, but also when words are created by compounding, reduplication, and other word-formation processes. For this reason, Boas-which was designed to serve a machine-translation system-treats derivational morphology in a special way. First, it elicits L affixes for an inventory of some 100 productive, generally compositional meanings that are realized affixally in many languages (e.g., negation (un-), opposition (anti-), and inexact

likeness (pseudo-)). Then it elicits all affixes that primarily change the part of speech (e.g., -ly in English makes adverbs out of adjectives: quick à quickly). Finally, it permits the user to list any other derivational processes "free form". That is, the informant types in some affix, indicates what part(s) of speech it attaches to, what part(s) of speech result, and what meaning the affix carries.

Derivational information represents an important descriptive aspect of language, but one that must be entered upon with caution due to the abovementioned pitfalls of non-compositional meaning. In LCTL-Boas, one could incorporate more descriptive power into the derivational-morphology module, since the resulting descriptions will not need to be automatically converted into processing rules. That is, rules that are helpful to people but would be too "loose" for machine processing could be elicited and stored as part of the grammar description.

Closed-Class Lexical Acquisition

The closed-class lexicon contains a finite inventory of cross-linguistically prevalent semantic meanings that include things like spatial relations; temporal relations; case relations; personal, reflexive, relative, interrogative, indefinite, predicative, demonstrative and possessive pronouns; conjunctions; articles; quantifiers; cardinal and ordinal numbers; and interrogative adjectives and adverbs.

From the cross-linguistic perspective, it is important to conceptualize the closed-class lexicon as meaning-oriented rather than part-of-speech oriented because the realization options for this collection of meanings reach beyond the familiar word and phrase options of the open-class. That is, closed-class meanings are also regularly realized as an affix or inflectional feature. For example, the English preposition the is translated by the Bulgarian suffixes -to, -ta, etc.: more 'sea' ~ moreto 'the sea'; the Persian possessive pronoun your can be translated by the suffix t: kt|b 'book' ~ kt|bt 'your book'; and the English reciprocal oneself can be translated by the Russian suffix -sja: myt' 'to wash' ~ myt'sja 'to wash oneself'. Feature realizations of closed-class meanings include the well-known use of the Instrumental case to indicate instrumental with: e.g., Polish rewolwerem, the Instrumental Singular of rewolwer 'revolver', can mean '(shoot, kill, etc.) with a revolver'.

Apart from extended realization options, there are other features that distinguish closed-class elements from open-class ones. First, if closed-class items inflect, they often require different paradigm templates than those found for the open-class parts of speech (e.g., pronouns tend to be singular only or plural only). Second, rules of inflection that apply to open-class elements may well not cover

closed-class elements, so a full listing of forms might be necessary or preferable.

The closed-class interface in Boas was designed to speed acquisition while providing for all possible L realizations of the English word senses. The look and feel of the interface is illustrated in Figure 4 using a portion of the temporal relations page, with Russian equivalents listed. (The coverage, interface functions, and user-oriented issues of the closed- and open-class lexicons are discussed in McShane and Zacharski 2003.)

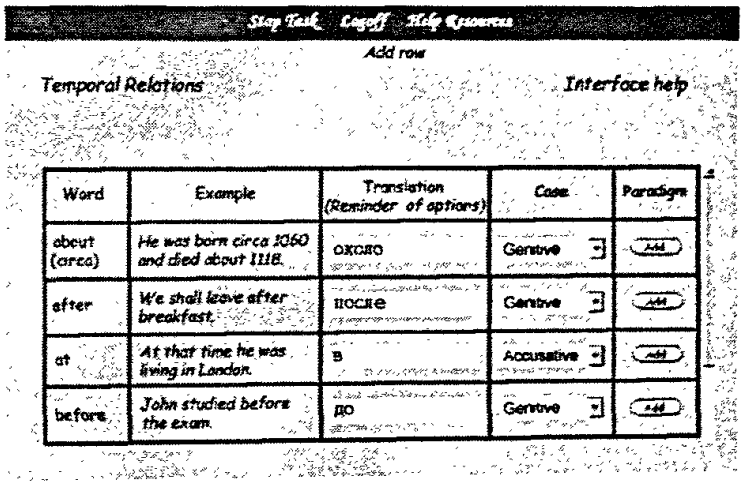


Figure 4

In LCTL-Boas, the elicitation of closed-class elements could remain exactly as in Boas but additional viewing options of the closed-class lexicon could be provided. Vocabulary exercises to drill closed-class meanings and forms could be similar to those for the open class, described below.

Open-Class Lexical Acquisition

The open-class lexicon contains words and phrases from the major parts of speech-nouns, verbs, adjectives and adverbs-plus proper nouns, adjectives derived from proper nouns, acronyms, abbreviations, set phrases and idioms. Figure 5 shows the basic lexical acquisition interface on the example of a system for Russian.

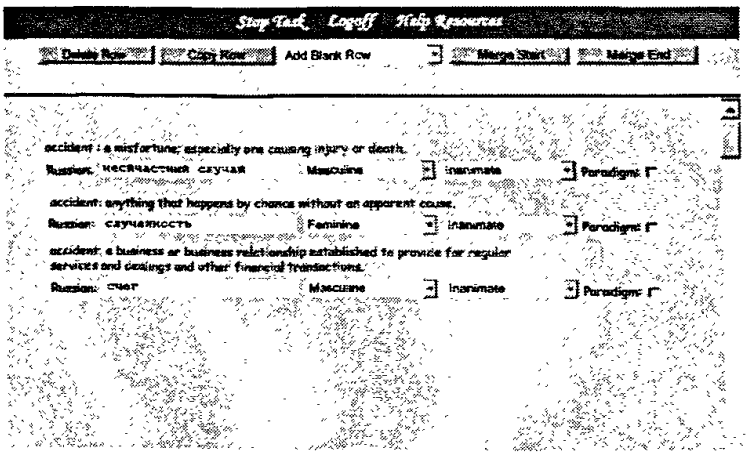


Figure 5

Since Boas is intended for languages for which few or no resources are available, the method of translating word lists is expected to dominate the acquisition process. English-driven acquisition using Boas's resident word lists is one option. Another acquisition option is for the informant to translate word lists that he or she compiles off-line. Such lists can be in L or English, can cover a specific subject area or be generalized, and can be gathered using Boas's corpus tools or any other means. Importation instructions are provided and include information about what will happen to duplicate entries, if any should occur. Working from externally generated lists is highly recommended, at least as a supplement, for languages with widespread derivational word-formation processes like compounding and reduplication. Most such forms will not have correlates in the English word lists used for acquisition, and rules for their creation are not specifically elicited in Boas because of frequent semantic non-compositionality in derived forms. The goal of presenting all of these options is to cater the acquisition process to the envisioned needs, resources, and preferences of the user. Such options will be particularly important for LCTL-Boas because teachers and/or students themselves can cater a system to specific needs of any given class, level, etc.

For flective languages, all "regular" words in the open-class lexicon can be analyzed using the rules generated during paradigm elicitation. All irregular forms must be listed explicitly. One enhancement that would benefit a pedagogical system would be for a teacher to be able to quickly check through the forms of each word that the rules would generate, resolve any ambiguities, and correct any errors. After each word was checked, all its forms - associated with its features (e.g., Genitive Singular) - would be saved and no longer be subject to analysis. Analysis would continue for those words

whose forms were not saved explicitly.

The most obvious direct pedagogical uses of the lexicons are to provide a reference for students and to promote vocabulary acquisition through drills. Drills could even be open-ended, incorporating knowledge resident in the profile of L: e.g., a drill could have students provide all the singular forms of random nouns from nominal Paradigm 4 (which might be Feminine nouns ending in -a) until they get 10 sets of correct answers, at which point the drill would be finished. Other types of on-line vocabulary drilling are well known and will not be reiterated here.

Syntax

We leave syntax until last because this module has the most visible trace of Boas's original use as a support for machine translation. Syntax has a special status in the machine processing of text. Despite vast attempts to write full and sufficient syntactic grammars of languages for machine processing, coverage is universally insufficient, which has led to a trend in "less syntax" (with no worse results) in natural language processing systems. As such, the syntactic phenomena elicited in Boas are not exhaustive. The system covers:

- the structure of a noun phrase (its components and their ordering),
- means of realizing grammatical functions (e.g., the subject can be realized by case-marking, position in the clause, etc.),
- means of realizing sentence types (imperatives, interrogatives, etc.), and
- a represenative but not complete inventory of syntactic constructions (e.g., affix hopping, topic fronting).

While this actually covers many of the most important aspects of syntax cross-linguistically, it does not cover all that one will need for teaching purposes. Expanding the syntactic portion of Boas for LCTL-Boas would not be difficult, however, because in a learning system, user input need not be automatically converted into processing rules, it can simply be saved like a document. Therefore, the only development challenge would be to collect a very large inventory of syntactic phenomena in order to remind the language informants about them, should they be relevant for L. Moreoever, since Boas is designed modularly, expansion of this or any module is is possible at any time.

More on Configurable Exercises

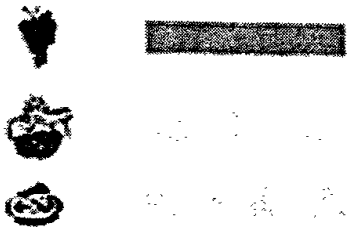
At least three classes of exercises could be incorporated into LCTL-Boas. All would be ramped-up with L information but each would require a different degree of superuser input. On one end of the spectrum are MOSTLY PREPARED EXERCISES that require only the L

responses to be entered-e.g., vocabulary quizzes using preselected pictures, as shown in Figure 6. Part I is the template for the superuser; part II is what a student will see after he or she has input answers and automatically received corrections by the system.

Template For Picture-Prompted Vocabulary Exercises

Name of exercise:
Instructions:
Creator of exercise:
Date:
Status (graded/ungraded):

INSTRUCTIONS FOR CREATING THE EXERCISE
Type in the names of only those food items you wish to be included in this exercise, leaving the other slots blank. If there is more than one acceptable variant, hit Return between variants.



etc.

Click here to import more pictures of food items.
Test exercise before committing it to system.
Commit exercise to the system at.... [search directories].

Assume the user already input Czech responses. The second column shows corrections.

Food, Basic: Answers

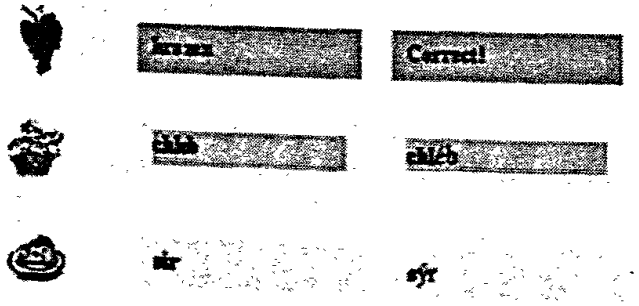


Figure 6. An example of mostly prepared exercises: first the teacher's template, then the students' version, with errors automatically corrected.

On the other end of the spectrum are FREE-FORM EXERCISES, for which the system supplies a template and the superuser inputs both the

questions and the responses. These can include multiple choice questions, fill-in-the-blanks exercises, question-answering based on an imported or typed text, etc. The range of material included in free-form exercises is open to limitless creativity. An example is shown in Figure 7.

Template For Free-Form Exercises

Name of exercise: Humpty Dumpty
Instructions: Fill in the blanks of the Humpty Dumpty nursery rhyme.
Creator of exercise: Dr. Jones
Date: 10/12/02
Status (graded/ungraded): graded

Click *here* to import a text from the web.
Click *here* for a textbox to compose a text.
Click *here* to create an exercise with multiple-choice responses.

Next screen for the superuser

	Type prompt. Place a star at point of answer insertion.	String Response. Hit "Return" between variants, if applicable.
1.	Humpty Dumpty sat on *.	a wall
2.	Humpty Dumpty had *.	a great fall
...		

Test exercise before committing it to system O
Commit exercise to the system at.... [search directories].

What the student will see.

Humpty Dumpty
(graded)

Fill in the blanks of the Humpty Dumpty nursery rhyme.

1. Humpty Dumpty sat on _____.
2. Humpty Dumpty had _____.
...

Save answers but do not submit yet. O
Submit exercise to Miss Jones. O

Figure 7. An example of free-form exercises: first the teacher's template, then the students' version, with errors automatically corrected.

In the middle of the spectrum lie partially prepared exercises—the type of exercises that are helpful when studying any language, but for which the majority of content must be supplied by the superuser. Idiom exercises, for example, fall into this category.

Template For Idiom Exercises

Name of exercise: Idioms, week #1
Instructions: Click on the correct rephrasing of each underlined idiom.
Creator of exercise: John Thatcher
Date: 5/3/02
Status (graded/ungraded): ungraded

Number	Type Prompt. Surround idiom by stars.	Responses. Click on the radio box next to the correct answer.
1.	"What did John get on the test?" "*You got me.*"	a. I forgot. b. It's not nice to ask such things. c. I don't know.
2.	"Mark says he makes 100 grand a year." "*Gimme a break!*"	a. He's lucky! b. That's not true! c. What a trivial fact!
...		

Test exercise before committing it to system. O
Commit exercise to the system at.... [search directories].

What the student will see.

Idioms, week 1
(Ungraded)

Click on the correct rephrasing of each underlined idiom. The text field will show "Correct!" or "Try again."

"What did John get on the test?" "You got me."
a. O I forgot.
b. O It's not nice to ask such things.
c. O I don't know.
etc.

Figure 8. An example of partially prepared exercises: first the teacher's template then the student version.

Exercises like these could be created to drill the information in any of the modules of Boas, as well as anything outside of Boas's scope. As mentioned earlier, the inventory of such exercises would reflect judgments by the teachers who would consult for LCTL-Boas regarding what is most helpful. Aspects of natural language processing and user modeling could be incorporated (e.g., automated checking of sentences freely produced by students, automated evaluation of student progress and determination of further learning tasks), but that would require research efforts above the development efforts discussed here.

More on the Task Tree

The same type of control structure used in Boas and the larger Expedition System could be used in LCTL-Boas with the same benefits: free ordering of tasks apart from prerequisites, redo capabilities, etc. Excerpts from the task tree for a language pedagogy system might look as follows, with the superuser and the student users being presented with different task inventories (Figures 9 and 10, respectively).

Teacher (Superuser) Main Menu
Create/expand the bilingual lexicon
  Create the bilingual lexicon from scratch with English prompts
  Import a bilingual lexicon
  Link to an on-line bilingual lexicon
  Import another user's lexicon
  Expand the current lexicon
Create/expand the morphological analyzer
  (An extensive process, as in Boas)
Create exercises
Create largely prepared exercise
Create partially prepared exercise
Create free-form exercise
Review/edit existing exercises
Post/retrieve/grade assignments
  Post a new assignment
  Retrieve and grade submitted assignments
Create/modify grade book
Post information to class
Post assignments
Post grades
Post other notes
Create Web links or reference materials

Figure 9

Student (User) Main Menu
Class Requirements
Do tonight's homework
  Check corrected previous homeworks
  Check syllabus
  Check grades
Practice/Study
Practice previous assignments
Practice supplementary exercises
Review graded homeworks and tests
  Create own exercises
Vocabulary/dictionary
Use/search the bilingual lexicon
  Expand the bilingual lexicon
E-mail
  Check class mailing list
  Send e-mail to the teacher or list
Web links

Figure 10

These task trees show one important but as yet not discussed aspect of a system like LCTL-Boas: it would provide the control structure not only for creating and using language materials, but also for all manner of course-organization functions, like communication between users, automatic tracking of assignments, etc. Each of these enhancements would require some development efforts, but the basic architecture is in place.

Summing Up

We believe that Boas, even in its current form - or modified slightly to include nicer formatting of the language profile - could be exploited by teachers of LCTLs to record, in an organized and system-guided fashion, the basic knowledge and lexis of LCTLs for distribution to students. However, even better would be to enhance this system, incorporating language-learning modules into the architecture to create a new LCTL-Boas. Development of these modules would be grounded in the same template orientation, knowledge elicitation methodologies, and user-support techniques as were designed for Boas. Specifically,

- Whereas typological knowledge about language in general grounded the form and content of Boas, knowledge about language pedagogy in general could ground the form and content of LCTL-Boas. That is, the teaching community knows what kinds of exercises and practice materials best target various aspects of language-learning, and these can be con-

verted into template form, with the content provided by the superuser for each language.

- Whereas KE methodologies lead language informants through the process of describing a language in Boas, similar KE methodologies can lead teachers through the process of creating and grading exercises on-line, and students through the process of doing and submitting those exercises in LCTL-Boas.
- Just as language profiles created through Boas can be shared over the Web, so could other materials developed by teachers through LCTL-Boas.
- The independent mode of using Boas promotes what Brecht and Walton call "learner-managed learning".

Just as Boas occupies a particular corner on the landscape of machine translation, expediting the elicitation of structured language knowledge to serve machine translation from rarer languages into English, so would LCTL-Boas fill a particular niche. Although it could be used to describe any natural language, including the commonly taught ones for which extensive paper and computer-based resources exist, it would most crucially serve languages lacking sufficient descriptive and/or computer resources. Whereas teachers of French or Spanish can choose from an array of teaching materials at all levels, selecting their favorite approach to grammar description, deciding how important they deem a CD-ROM or video supplement, etc., teachers of LCTLs tend to have zero or one book to "choose from", and if they don't like the approach or presentation of material they must create their own. LCTL-Boas would support this effort in the many ways detailed above, as well as permit the sharing of language profiles over the Web such that the teaching community for a given LCTL could benefit from each others' efforts. Thus, LCTL-Boas's tools for creating a language profile lie outside of the well-known debate "do computers really expedite language learning?" Having language-learning materials catered to a teacher's preferences and needs will certainly promote at least the teaching end of language pedagogy. How those materials are later used in the teaching process - e.g., which (if any) on-line drills will be engaged in by students - is another question.

The efficacy of drilling grammatical forms has been widely disputed, as noted by Armstrong and Yetter-Vassot (1994, 477): "Many believe that learning and practicing the linguistic rules of the foreign language has very little impact on the speaker's ability to produce grammatically appropriate utterances. On the other hand, there are those who believe that students need to spend a certain amount of time practicing with forms in order to improve the accuracy of the message." It is hard to imagine how the latter point could be

questioned: if a student cannot remember that the Accusative case form of the Russian noun ruka 'arm' is ruku, he or she will never be able to construct a grammatical sentence using this word as a direct object.

Technology-supported language learning has recently reemerged after a couple of decades of lagging interest following rather unsuccessful experiments in the language laboratory in the 1950's and 60's. Schwartz (1995) frames this history as a cautionary tale, reminding us that technology must be used creatively if we expect its benefits to exceed those gained through traditional methods. Embellishing his idea, I would suggest that the goal is not simply to transport what is done in the classroom to a lifeless computer terminal but to rethink the "givens" of language pedagogy using inspiration derived from, and methods supported by, technological advances. In doing so, however, we must keep reasonable expectations of what computers can presently do and what they can be expected to do in the foreseeable future, not waiting for research in artificial intelligence to produce a teacher in a laptop.

It is interesting to note that the sections on artificial intelligence and computational linguistics in *Computational Applications in Second Language Acquisition* (32-36) focus in large part on developing dialogue systems, with the computer being expected to understand the faulty input of the student and generate a flawless response or correction using natural language. While research in this area is fascinating and both draws from and contributes to other natural language processing tasks (e.g., question answering, machine translation, text summarization), carrying on a free-form conversation, especially with imperfect input, is one of the most daunting of all natural language processing tasks. Thus, in the short term, incorporating less ambitious NLP support for language learning, as would be done by LCTL-Boas, seems optimal. The importance of approaching computer-aided instruction with human-driven creativity rather than in hopes of computer-offered quick fixes cannot be overstated. As Armstrong and Yetter-Vassot (1994, 476) say: "If technology is to revolutionize language teaching, then instructors must be willing to invest time and energy in developing creative and pedagogically sound activities that will utilize those technologies..." The modules of LCTL-Boas could be used to any extent and in any combination. For example, a teacher could create a profile of L and distribute it to the students as a grammar (with lexicon), never pursuing any on-line drilling, testing, or class-organization functions. Or several teachers could cooperatively produce a profile as a summer project then cooperatively or separately expand those profiles throughout the school year, adding exercises, drills, and

vocabulary as the course developed. Or a teacher could select to use the system only for vocabulary-oriented drilling, circumventing the need to create a full language profile. Or a highly motivated student could study a language independently, creating a profile using available print resources and input from native-speaker informants, and having an expert in that language (living in any corner of the world) evaluate the results at regular intervals over the Web. In short, just as LCTL-Boas would be modular, so would its employment be open to mixing and matching functionalities for a given classroom, or extra-classroom, situation.

Boas has many advantages as a potential anchor for large-scale development of a suite knowledge-based language-learning tools, not the least of which is that it already exists. However, the methodologies, approaches and reasoning for them described in this paper are not bound to a given implementation of a given system but, rather, are intended to act as a suggestion to the teaching community regarding possible ways of exploiting technology, linguistics, and knowledge-elicitation strategies to create much needed language resources.

In short, we believe that the union of a language profile created in template-like fashion with learning resources developed using a similar approach could create a powerful resource for the advancement of teaching LCTLs.◆

References

Armstrong, Kimberly M. and Cindy Yetter-Vassot. 1994. Transforming teaching through technology. *Foreign Language Annals* 4(27): 475-486.

Brecht, R. D. and A. R. Walton (No Date). *National strategic planning in the less commonly taught languages.* Available at from World Wide Web:http://www.councilnet.org/pages/CNet_VLib_Pubs.html#Council papers.

Chapelle, Carol A. 2001. *Computer Applications in Second Language Acquisition.* Cambridge: Cambridge University Press.

Last, R.W. 1989. *Artificial Intelligence Techniques in Language Learning.* New York: Halsted Press.

McShane, Marjorie. 2003. Redefining "paradigm" for (computer-aided) language instruction. Forthcoming in *Foreign Language Annals.*

McShane, Marjorie and Nirenburg, Sergei. 2003a. Blasting open a choice space: Learning inflectional morphology for NLP. Forthcoming in *Computational Intelligence.*

McShane, Marjorie and Nirenburg, Sergei. 2003b. Parametrizing, eliciting and processing text elements across languages. Under review at *Machine Translation.*

McShane, Marjorie, Sergei Nirenburg, Jim Cowie and Ron Zacharski. 2003. Nesting MT in a linguistic knowledge elicitation system. Forthcoming in *Machine Translation.*

McShane, M. and R. Zacharski. 2003. User-extensible on-line lexicons for language learning. Under review at *CALICO.*

Oflazer, K., S. Nirenburg and M. McShane. 2001. Bootstrapping morphological analyzers by combining human elicitation and machine learning. *Computational Linguistics,* 27(1): 59-85.

Schwartz, Michael. 1995. Computers and the language laboratory: Learning from history. *Foreign Language Annals* 28(4): 527-535.

Marjorie McShane is a Research Assistant Professor at the Department of Electrical Engineering and Computer Science of the University of Maryland Baltimore County.